Reference No. 4

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 940 754 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
08.09.1999 Bulletin 1999/36

(51) Int. Cl.^{6}: G06F 11/34

(21) Application number: 98119545.6

(22) Date of filing: 15.10.1998

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 02.03.1998 US 33040

(71) Applicant:
HEWLETT-PACKARD COMPANY
Palo Alto, California 94304-1181 (US)

(72) Inventors:
- Smith, Mark D.
  Fort Collins, CO 80525 (US)
- Ramanathan, Srinivas
  Sunnyvale, CA 94086 (US)
- Caswell, Deborah L.
  Santa Clara, CA 95051 (US)

(74) Representative:
Schoppe, Fritz, Dipl.-Ing.
Schoppe & Zimmermann
Patentanwälte
Postfach 71 08 67
81458 München (DE)

(54) Modeling of internet services

(57) The dependancies of a computer service are modeled. The modeling hierarchically defines the relationships between the computer service and the hardware and software services which the computer service depends. These relationships may be contained in data structures defining a directed acyclic graph. The model also defines which measurements need to be taken to determine health and performance of the computer service and the health and performance of all the computer services upon which the computer service depends. Software agents that take these measurements may be deployed using the model to determine the measurement locations and functions. Data from measurement agents may be propagated up the model hierarchy. The model (200) may also be visualized by a graphical interface to communicate the dependancies and the health and status of the services upon which the modeled service depends.

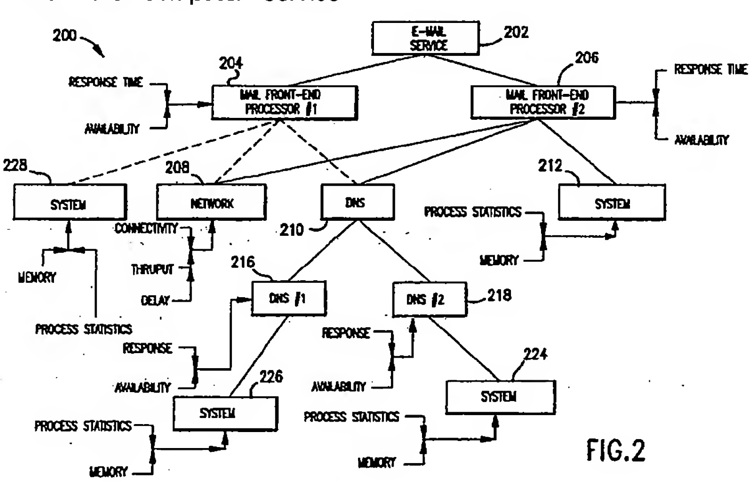

FIG.2

EP 0 940 754 A1

Description

Field of the Invention

[0001] The present invention relates generally to detecting and diagnosing problems with computer services. More specifically, the present invention relates to the autonomous collection, organization, distillation, and presentation of measurement data from computer services to enable operators to detect, isolate, and resolve faults and quality of service problems, as well as conduct service and capacity planning.

Background of the Invention

[0002] The worldwide network of computers commonly known as the "Internet" has seen explosive growth in the last several years. Mainly, this growth has been fueled by the introduction and widespread use of so-called "web" browsers, which allow for simple graphical user interface (GUI)-based access to network services such as E-mail, news, file transfer protocol (ftp), web pages, etc. Many people contract with an internet service provider (ISP) to obtain access to the Internet. Subscribers to an ISP typically use a personal computer and modem to connect to the ISP using the public switched telephone network. Once connected, the user may perform the desired functions.

[0003] In addition to providing a connection to the Internet, ISPs or other computer access providers (CAPs), such as corporate IT departments, often provide additional services that expand, enhance, or improve internet functions. For example, many CAPs provide users with the ability to send and receive E-mail. Or, the CAP may provide a local domain name server (DNS) to speed the resolution of the domain names the subscriber is trying to access, thereby improving overall access speed.

[0004] The customers of these services tend to view the quality of that service in simple terms: accessibility and performance (i.e. speed, responsiveness, etc.) Unfortunately, the accessibility and performance of a service may depend on many factors. First, there is the service itself and the servers that implement the service. These servers may be comprised of the server software, the hardware running the server software, the operating system running on the hardware, and the network hardware and software that support the implementation of that service. Finally, the performance and accessibility of the server components may further depend on other services, hardware, software, etc. For example, the quality of E-mail service may first depend on the hardware and software running the E-mail program. This E-mail program may depend on a DNS server and a network router. The DNS may be used to resolve domain names before the E-mail can be sent, and the router may be used to relay the E-mail from the CAP's local network to the Internet backbone. Finally, the performance of the DNS server may depend on the performance of a network file system (NFS) server and several other pieces of hardware, software, or services provided by the same, or different hardware and software. Each of the components that contribute to the performance of the E-mail service are interrelated and may be located on the same or different networks or hardware, rely on the same or different software and operating systems, or be running on the same, or different, hardware.

[0005] It can be seen from the previous discussion that the simple quality of service measures of accessibility and performance may depend on the interrelationships of many hardware and software components arranged in a complex system infrastructure. It is also likely that individual CAPs will have an infrastructure comprised of a unique arrangement of components and their interrelationships. This makes it difficult to construct a "one size fits all" solution to conduct service and capacity planning and to detect, isolate, and resolve faults and quality of service problems.

[0006] Many CAPs manage their networks and services on a rather ad hoc basis. Collections of management scripts available in the public domain and policies and procedures developed on the fly combine to provide what little proactive measurement and monitoring of the infrastructure there is. Detailed knowledge of the infrastructure, relationships, test and measurement techniques, policies and procedures are often passed around the CAPs staff by word of mouth. Relationships between all the infrastructural components are usually only understood by the most senior technical operations staff. Finally, changes in operational procedures and policies are usually initiated only after hard won experience in dealing with failures and quality of service problems have been internalized by the operations staff. This period of internalization with its associated failures and poor service can adversely affect a CAPs reputation, and cost a CAP customers, market share, and revenue.

[0007] Accordingly, there is a need in the art for a system that captures the knowledge and experience of the senior technical operations staff and make that information available to a much wider audience. Such a system should to be able to gather data from a variety sources and tools that test infrastructure elements, collect data from SNMP MIBs and log files and correlate that data into the information needed to enable less skilled member of the operations staff to detect, isolate, and resolve faults and quality of service problems. There is a need in the art for a system that includes the detection of potential problems before they create a failure, are detected by users, or result in a quality of service problem. There is a need in the art for a system that allows less skilled members of the operations staff to diagnose, isolate, and resolve failures and quality of service problems without consulting the senior technical operations staff.

Finally, it would be desirable if such a system could configure itself automatically and deploy the tools and test elements necessary for problem detection, isolation, and resolution.

SUMMARY OF THE INVENTION

[0008] These and other needs are satisfied by the methods and apparatus of the present invention for modeling, displaying, and gathering information relating to computer services. Working from configuration information, a hierarchical model of a computer service is constructed. This model is an explicit model of a computer based service. The model defines the structure and dependencies of a service and its components, the measurements, baselines and thresholds, the health status, alarms, and control of the service components. The model enables: easier fault detection and isolation, auto deployment and configuration of the system, dissemination of expertise, root cause analysis, and visualization of the structure and dependencies of a service and its components.

[0009] In one embodiment, this model may be represented as an acyclic graph. At the root of the model is the service itself. The next level is the servers that implement the service itself. Each of these servers is in turn are comprised of the server software, operating system, network interfaces, other services, etc. that support the implementation of that service. Further elements of the infrastructure are tied into the model at each level of the hierarchy. Finally, at the leaf nodes of this service model are the actual measurements that monitor fundamental aspects of the health and performance of each of the infrastructure components represented higher up in the hierarchy. The hierarchical model establishes how each service depends upon other services, software, hardware, and networks. A component of a service may have its own model. The model of this component may be instantiated in the model of the service. These models, and the functions they help acheive, may be implemented by data structures and programs running on one or more computer systems.

[0010] A representation of the hierarchical model provides an easy to understand method of viewing the complex relationships of infrastructure elements and the measurements that indicate the health of each modeled infrastructure element. This representation may be displayed using a graphical user interface (GUI). Measurements, as well as an indication of the health of each infrastructure may be displayed on the representation of the model.

[0011] The hierarchical model also provides a template for automatically deploying software agents to take measurements of the fundamental aspects that affect the health and performance of the service. These measurements are propagated back up the hierarchical model to provide an indication of the overall health and performance of the service. Individual measurements may be detected and deemed abnormal as defined by deviations from baselines and/or threshold values. Abnormalities due to the cumulative effects of several dependant infrastructure elements may also be detected by propagating measurement information up the model hierarchy and then applying an arithmetic, or alternatively, a fuzzy-logic test. When abnormalities are detected, the model may also contain control definitions. These control definitions may define what actions to take to resolve the abnormality.

[0012] Problems with computer services may be diagnosed using the dependancies established by the hierarchical model by descending the hierarchy of the problem service. As the model is traversed from the problem service to lower levels of the hierarchy, it is examined for other services that are having problems. Healthy elements, and the services they depend on, are quickly eliminated. Root cause determination is facilitated by narrowing the search to only a few elements. This search may be conducted automatically, or with the aid of a GUI displaying a representation of the model, indications of the health of each infrastructure component, and measurement data.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013]

FIG. 1 is a schematic illustration of a representative CAP.
FIG. 2 is a schematic illustration of part of a hierarchical model of an E-mail service of FIG. 1.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0014] FIG. 1 shows a schematic illustration of a computer access provider (CAP). Users connect via the telephone network to one of the modems **112** that are connected to a modem server **110**. Modem server **110** connects to at least one local area network **120**. This network **120** allows communication between other computers (**102, 104, 106, 108, 128, and 114**) within the CAP that are also connected to the network **120**. The network is also connected, via some means, to the Internet **126**. In this representative diagram, that connection is shown as being via a router **116**, a CSU/DSU **118**, and a leased line **124**.

[0015] Services may be provided by software running on computers (**102, 104, 106, 108, 110, 128**, and **114**) connected to network **120**. In FIG. 1, E-mail is provided by processes running on computers **106** and **108**. The CAP of FIG.

1 is configured so that requests for E-mail services are divided between computers **106** and **108** Domain name service (DNS) is provided by processes running on computers **104** and **128**. USENET news service is provided by processes running on computer **102**. Finally, computer **114** is used to provide overall administration.

[0016] FIG. 2 illustrates a hierarchical model **200** of a E-mail service of the representative CAP of FIG. 1. At the root of the model **200** is the E-mail service represented by box **202**. As discussed above, the E-mail service of the CAP of FIG. 1 depends upon two E-mail front-end processes running on two separate computers **106, 108**. These processes form the next level of the hierarchy and are represented as boxes **204** and **206**. The dependancy of the E-mail system on these two front-end processors is shown by the solid lines running from box **202** to boxes **204** and **206**. The measurements that are indicative of the performance of these processes are response time and availability. These leaf nodes of the model are shown as arrows feeding into boxes **204** and **206**.

[0017] The performance of E-mail front end processor #2 as represented by box **206** depends on the performance of: the network, the DNS, and computer system **108**. These are represented, in order, by boxes **208, 210,** and **212**. The dependancy of E-mail front end processor #2 on these services is shown by the solid lines running from box **206** to boxes **208, 210,** and **212**. The performance of E-mail front end processor #1 as represented by box **204** depends on the same network and DNS as E-mail front end processor #2, but a different computer system. These dependancies are shown by the dotted lines from box **204** to boxes **208, 210,** and **228**. The performance of E-mail front end processor #1 depends on the performance of computer system **106** whereas the performance of E-mail front end processor #2 depends on computer system **108** so they have different connectivity in the model.

[0018] The measurements that are indicative of the performance of the computer systems represented by boxes **228** and **212** are memory and process statistics. These leaf nodes of the model are shown as arrows feeding into boxes **228** and **212,** respectively. The measurements indicative of the performance of the network represented by box **208** are connectivity, thruput, and delay. Connectivity measures whether certain connections can be made, or whether certain computers may be reached, thruput measures how much data per second can be transferred across the network, and delay measures how long it takes data to get to its destination. These leaf nodes of the model are shown as arrows feeding into box **208**. The DNS service as represented by box **210** depends on the performance of two DNS processes running on two separate computers **104, 128**. These processes form the next level of the hierarchy and are represented as boxes **216** and **213**. The dependancy of the DNS service on these two processes is shown by the solid lines running from box **210** to boxes **216** and **218**. The measurements that are indicative of the performance of these processes are response time and availability. These leaf nodes of the model are shown as arrows feeding into boxes **216** and **218,** respectively.

[0019] The DNS #1 and DNS #2 processes depend on the performance of the computer systems they are running on: computer system **104** for DNS #1 and computer system **128** for DNS #2. These computer systems are represented by boxes **226** and **224**, respectively. The dependancy of DNS #1 on computer system **104** is represented by the solid line running from box **216** to box **226**. The dependancy of DNS #2 on computer system **128** is represented by the solid line running from box **218** to box **224**. The measurements that are indicative of the performance of the computer systems represented by boxes **226** and **224** are memory and process statistics. These leaf nodes of the model are shown as arrows feeding into boxes **226** and **224,** respectively.

[0020] The advantages of this modeling in deploying measurement agents can been seen by examining FIG. 2. Agents to monitor the health of the E-mail system may be deployed simply by traversing the model and starting agents, on the appropriate computer systems, to take the measurements indicated by the leaf nodes. By deploying only those agents indicated by the leaf nodes of the model, unnecessary agents are not deployed, unnecessary measurements are not taken, and the whole process can be automated using a tree-traversing algorithm. Because two or more high-level services may depend upon the same low-level service, the possibility of redundant agents being deployed exists. This is not necessarily bad. However, a simple method for avoiding this, if desired, would be to mark leaf nodes as they are deployed, and then not deploy a measurement agent for marked leaf nodes.

[0021] Once the measurement agents are deployed, the model may also be used to notify the operations staff of potential problems. Take the case where computer system **128** is about to run out of memory. This would be detected by the memory measurement agent represented by the leaf node MEMORY feeding into box **224**. The memory measurement agent would realize that the amount of memory used has exceeded a pre-set threshold. It would then relay this information, possibly in the form of a alarm, to a system containing the model. In FIG. 1 this would most likely be computer system **114**. Computer system **114** could then look at the model and determine, by following the model dependacies up the model, that DNS #2 was about to have a problem, and that may cause overall DNS problems. It could then determine that a problem with the DNS could cause problems with both E-mail front end processors which could cause problems with the E-mail service. It could make these determinations by traversing the model from box **224** to box **218** to **box 210** and then to each of boxes **204** and **206,** in turn, which both lead to box **202**, representing E-mail service. Therefore, using the model, it can be determined which services are going to be affected by a potential, or real, problem.

[0022] These problems, as well as all measurement data could be communicated to operations staff via a graphical

display. This graphical display could illustrate the dependancies of the infrastructure components by presenting a visual representation of the model much like FIG. 2. Measurement data could be illustrated as leaf nodes feeding infrastructure components, or as bar graphs, dials, and other indicia inside the representations of individual infrastructure components. Problems, potential problems, and measurements that have exceeded thresholds could be emphasized on the display by flashing or changing the color of the representation of that service or measurement.

[0023] To construct the hierarchical model, the information for that service must first be defined. One way to do this would be via a GUI where icons are used to represent services and lines, or links, can be drawn to establish dependancies. A simpler way would be to use a text file. In the case of a text file an entry for each service is created that assigns a name to the service, chooses the type of service from a group of predefined services, and then optionally specifies the components that this service is dependant upon and also optionally measurements to be taken that are indicative of the performance of that service, and parameters necessary for those measurements. Entries that define measurements, how to take them, and the baselines and thresholds for that measurement that indicate a problem could also be included. These entries may define, or re-define templates and threshold values. A portion of a sample file is shown in table 1.

[0024] Table 1 shows entries for the E-mail service, mail front-end processor #1, and the computer system that runs mail front-end processor #1. These services correspond to boxes **202, 204,** and **228** in FIG. 2, respectively. Table 1 also contains and entry that could be used to override the default threshold for the measurement of CPU usage.

Table 1.

begin-service	
name	= Mail-Service
type	= STServiceMail
components	= Mail-Front-End-1, Mail-Front-End-2
measurements	= Mail-Response
end-service	
begin-service	
name	= Mail-Front-End-1
type	= STServiceMail
components	= Mail-Front-End-1-Host
measurements	= Mail-Response(localhost, testmachine.com), Mail-Availability(testmachine.com)
end-service	
begin-service	
name	= Mail-Front-End-1-Host
type	= STHost
measurements	= CPU-Stats(mailhost1.hp.com), MEM-Stats(mailhost1.hp.com)
end-service	
begin-measurement	
name	= CPU-Stats
script	= rsh $p1 load --percent
type	= gauge
unit	= percent
threshold	= 150
baseline	= standard

Table 1. (continued)

end-measurement	

[0025] The measurements field in Table 1 specifies the measurements that are to be taken to establish an indication of the health of the service. For example, the MEM-Stats measurement may run a script on the machine specified as a parameter that returns the amount of free memory on that machine. This return value would be an indication of whether more memory, or virtual memory, needs to be added to that machine.

[0026] A fairly simple parsing algorithm could be used to read the configuration information and construct a model of the service. The model could be comprised of one or more directed graphs stored in computer memory using any number of conventional data structures. The measurements could be processing running on local or remote machines, but return their data to a supervisory program that stores measurement values in the model. The supervisory, or another program, could be used to propagate alarm conditions around the model according to the dependancies specified by the model. The supervisory, or another, program could read the model for dependancy information, measurement values, alarm conditions to provide a graphical display of the status of a service and all of its infrastructure components. Programs that read the model could also be used to diagnose and isolate problems by allowing a user to traverse the model looking for problems conditions while the model continually received updates from the measurement processes.

[0027] It will be appreciated that the instant specification is set forth by way of illustration and not limitation, and that various modifications and changes may be made without departing from the spirit and scope of the invention.

Claims

1. A method of modeling a computer service (202), comprising:

(a) defining a first set of services that said computer service depends upon, wherein said first set of services has at least one member; and,
(b) constructing a model (200) of the relationships between the members of said first set of services and said computer service.

2. The method of claim 1, wherein there is a first set of measurements for each member of said first set of services that give an indication of the performance of that service and wherein the members of said first set of measurements and said model is used to determine which member of said first set of services is causing abnormal performance of said computer service.

3. The method of claim 1, wherein there is a first set of measurements for each member of said first set of services that give and indication of the performance of that service and wherein said model is used to deploy a measurement agent that takes a measurement that is a member of said first set of measurements.

4. A model (200) of a computer service (202), comprising:

a first data structure containing the relationship between said computer service and a first service, wherein the operation of said computer service depends on said first service; and,
a second data structure containing the relationship between a first measurement and said first service.

5. A model (200) of a computer service (202), comprising:

a plurality of a first data structure, wherein said first data structures contain the relationships between said computer service and a plurality of services, wherein the operation of said computer service depends on said plurality of services, and wherein said first data structures also contain the relationship between a plurality of measurements and said plurality of services, and wherein each of said plurality of measurements indicate the performance of at least one of said plurality of services.
6. The model of claim 5, wherein said model defines a set of measurement agents that are executed to take each of said plurality of measurements.

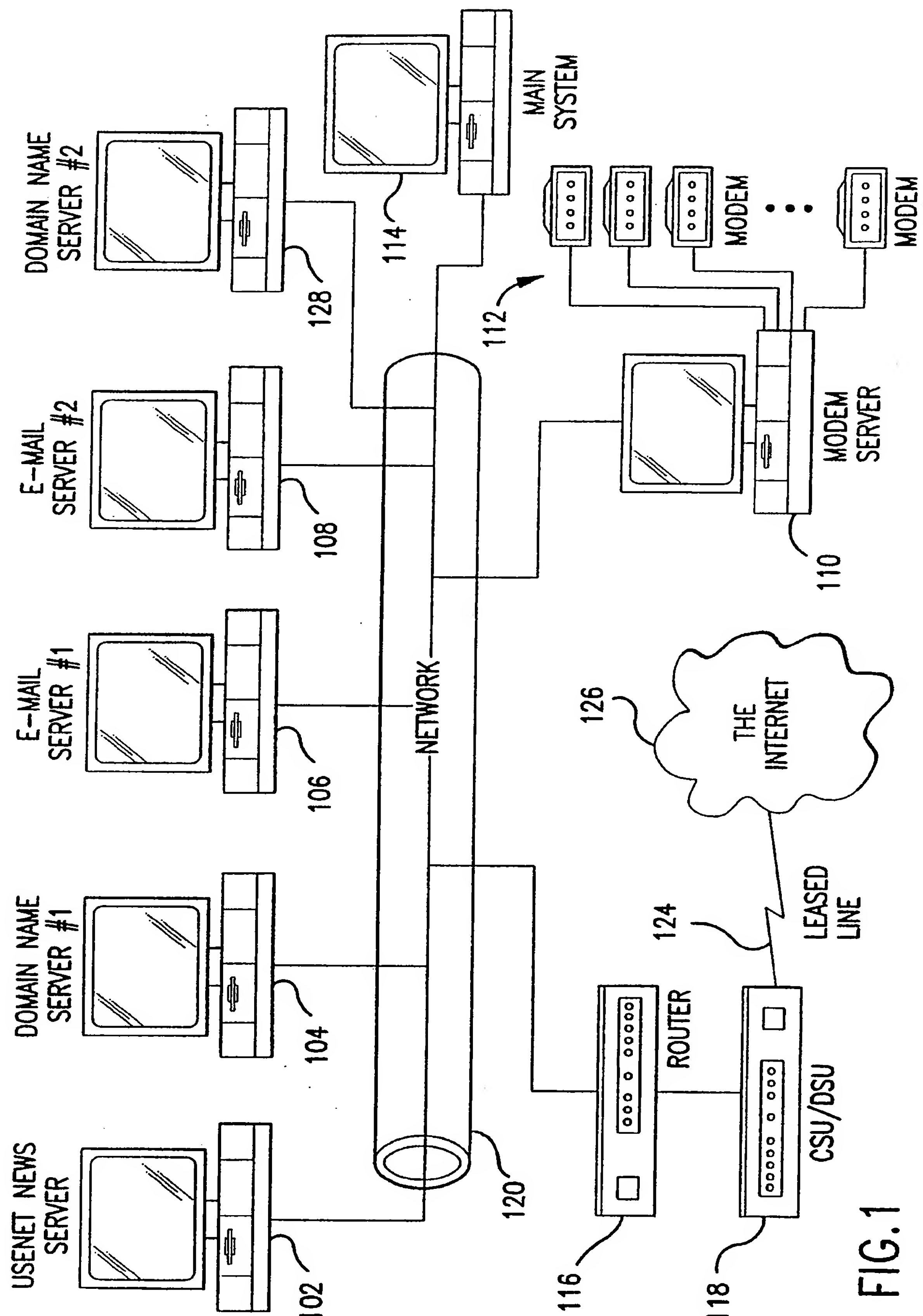
DOMAIN NAME SERVER #2
128
MAIN SYSTEM
114
112
MODEM
MODEM
E-MAIL SERVER #2
108
MODEM SERVER
110
E-MAIL SERVER #1
106
NETWORK
126
THE INTERNET
DOMAIN NAME SERVER #1
104
124
LEASED LINE
ROUTER
USENET NEWS SERVER
102
120
116
CSU/DSU
118

FIG.1

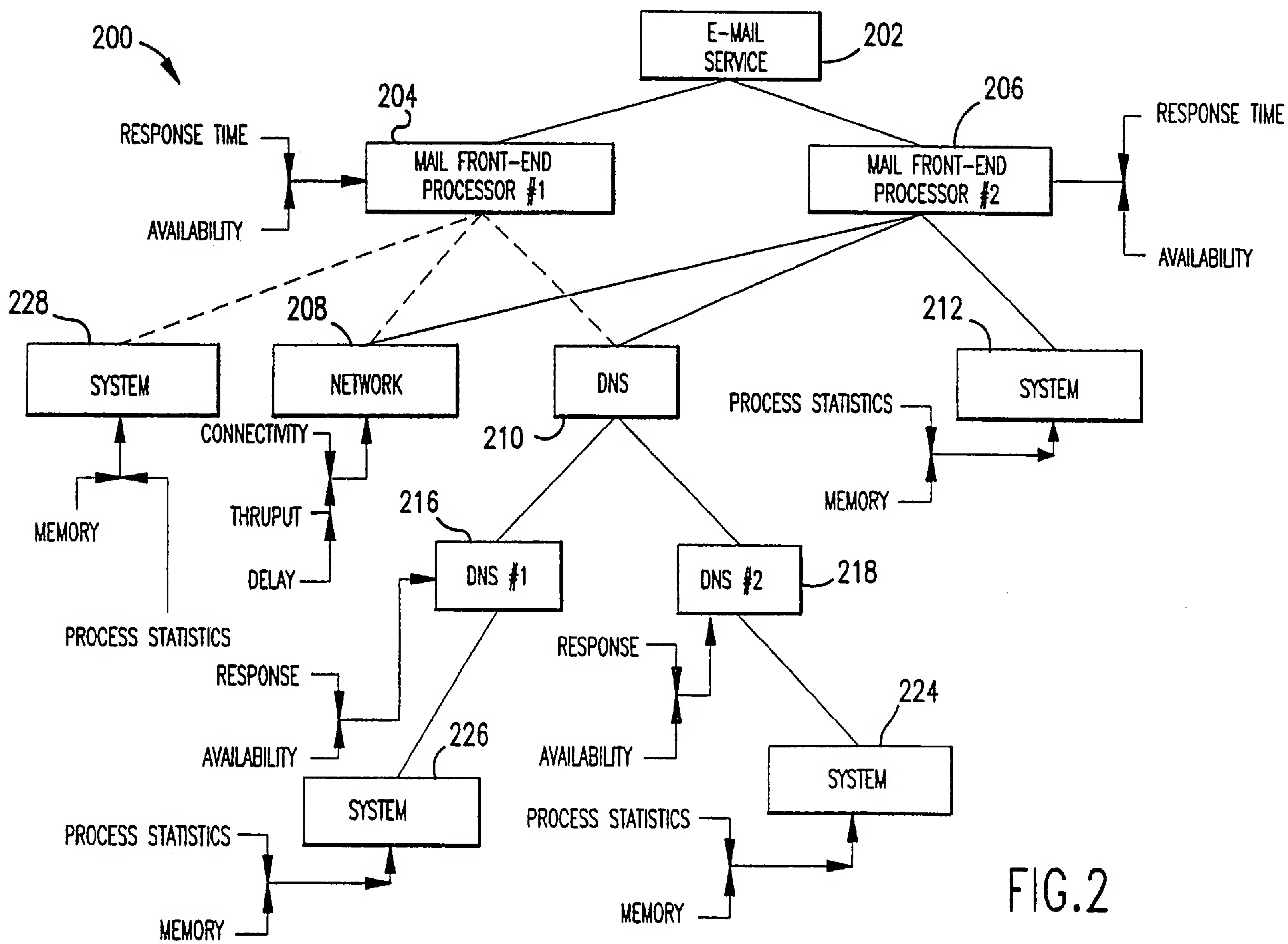
200
E-MAIL SERVICE
202
204
MAIL FRONT-END PROCESSOR #1
206
MAIL FRONT-END PROCESSOR #2
RESPONSE TIME
AVAILABILITY
RESPONSE TIME
AVAILABILITY
228
SYSTEM
208
NETWORK
DNS
210
212
SYSTEM
MEMORY
PROCESS STATISTICS
CONNECTIVITY
THRUPUT
DELAY
PROCESS STATISTICS
MEMORY
216
DNS #1
DNS #2
218
RESPONSE
AVAILABILITY
RESPONSE
AVAILABILITY
226
SYSTEM
224
SYSTEM
PROCESS STATISTICS
MEMORY
PROCESS STATISTICS
MEMORY

FIG.2

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 98 11 9545

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X	Syllogic White Paper: "Adaptive Systems Management", Version 2.1, January 1998, pages 1-14. Retrieved from Internet: <URL: http://www.syllogic.com/aboutsyllogic/asm.html> 2 June 1999 XP002105194 * the whole document *	1-6	G06F11/34
X	US 5 276 877 A (FRIEDRICH ET AL.) 4 January 1994 * column 3, line 14 - line 39 * * column 6, line 20 - column 7, line 2 * * column 18, line 40 - column 19, line 7; figure 14 *	1	

TECHNICAL FIELDS SEARCHED (Int.Cl.6)

G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	10 June 1999	Herreman, G

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 98 11 9545

This annex lists the patent family membersrelating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

10-06-1999

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5276877 A	04-01-1994	CA 2047573 A	18-04-1992
		DE 4134419 A	30-04-1992
		FR 2668271 A	24-04-1992
		GB 2250111 A	27-05-1992
		JP 4299414 A	22-10-1992

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82